# Invoking Web Services based on Energy Consumption Models


Apostolos Papageorgiou, Ulrich Lampe,
Dieter Schuller, Ralf Steinmetz

*Technische Universität Darmstadt*
*Multimedia Communications Lab – KOM*
*Darmstadt, Germany*
*{name.surname}@kom.tu-darmstadt.de*

Athanasios Bamis

*University of Connecticut*
*Department of Computer Science and Engineering*
*Storrs, Connecticut, USA*
*bamis@engr.uconn.edu*



***Abstract*—Web service consumption may account for a non-negligible share of the energy that is consumed by mobile applications. Unawareness of the energy consumption characteristics of Web service-based applications during development may cause the battery of devices, e.g., smartphones, to run out more frequently. Compared to related exeprimental energy consumption studies, the work at hand is the first work that focuses on factors which are specific to *services computing*, such as the timing of Web service invocations and the Web service response caching logic. Further, Web service invocations are the only variable energy-consuming activity included in the experiments. Based on the results, it is shown, firstly, how the execution of exactly the same Web service invocations may lead to energy consumption results that present differences of up to ca. 15% for WLAN and ca. 60% for UMTS connections, and, secondly, how rules and techniques for energy-efficient development of mobile Web service-based applications can be extracted from the gained knowledge.**



***Keywords*-Web Services; Energy; Battery; Mobile;**


## I. INTRODUCTION

Enhancing Quality of Service (QoS) for Web services (WS) [1], usually in terms of performance of service invocations, has been established as one of the most important research goals ([2], [3]) in the domain of Service-Oriented Computing (SOC). The issue becomes even more crucial when mobile service consumers are involved, because the latter usually have limited resources in terms of processing power, memory, and battery. The goals of the QoS enhancements usually relate to parameters such as response time, user-perceived latency, execution time, or availability, which are also referred to as QoS parameters. Due to the increasing usage of Web services in mobile systems ([4], [5]) and the growing energy-related considerations, another performance-related aspect has been gaining in importance, namely energy (or battery) consumption [6].

The *energy consumption* caused by Web service invocations has rarely been considered as a Web service QoS parameter (cf., for example, [1], [7]), mainly because it is not a characteristic of the Web service alone, but it rather depends on a combination of factors. A further difference compared to other QoS parameters is that it is only coarsely and implicitly perceived by the user. However, no matter if energy consumption is considered to be a QoS parameter or not, it has many similarities with standard QoS parameters. For instance, it is heavily dependent on the amount of transmitted/processed data.

Indeed, for reasons that become clear by studying the energy consumption sources of mobile devices (cf. Section II), argumentations such as "the data transmitted wirelessly during Web service invocation have been reduced, thus energy is being saved" have been heard much more often than energy measurements have been presented. Thus, performance-efficient approaches that enhance other QoS parameters (e.g., response times) may have been thought to be energy-efficient, as well. However, recent research ([8], [9]) has indicated that other network- or application-related aspects may sometimes dominate over the amount of wirelessly transmitted data, so that more sophisticated and specialized approaches would probably deliver better results, if energy-efficiency is the primary goal.

The question that remains unanswered is to what extent the known factors (amount of transmitted data, device type, network connection type) *do* play an important role when it comes to Web service call sequences and if there are influences from any other factors, namely SOC-specific ones. These influences should then be analyzed, quantified, described and exploited.

The goal of the work at hand is to examine the influence of SOC-specific factors on the energy consumption of service invocations. For this, the notion of "service invocation patterns" will be introduced in order to design a series of experiments that focus, for the first time, on these SOC-specific factors. Then, based on the results, examples of defining or extracting rules for "energy-efficient Web service usage" will be provided in an attempt to pave the way towards the development of energy-aware Web service-based mobile applications.

## II. RELATED WORK AND CONTRIBUTIONS

A fundamental step is the analysis of the energy consuming (hardware) sources of the devices. Knowing which device modules dominate energy consumption is critical towards developing energy-efficient applications of any kind. In this direction, Zhang et al. [10] and Carroll et al. [11] explain how the display of a smartphone can have a significant share, usually varying in the range 0–0.5 Watt, depending linearly on the brightness level. The processor and memory modules also belong to the main energy consuming modules, reaching similar levels as the display and being lineraly related to the load and cubically related to the frequency [12], [10]. Even higher power consumption levels are reached by the wireless interfaces [6], e.g., WiFi, 3G, or bluetooth. Perrucci et al. [9] and Balasubramanian et al. [8] present relevant analyses, showing, among others, that WiFi is more energy-efficient than 3G, while 3G does not only have higher power peaks during transmission (usually ca. 2 Watt vs. ca. 1 Watt) but it also consumes energy in periods in which no data transmission ocurrs. Further smartphone energy consumption sources are the sensor modules (GPS, accelerometer etc.) and the audio modules.

*However*, as the research towards case-specific (and not only module-specific) analysis indicates that the findings of the referenced works are *not* enough. The exact way in which different domains (software technologies or usage scenarios) use and combine the mentioned modules also determines energy consumption. Thus, new models and rules are needed per domain.

Related approaches have appeared, for example, in the domains of e-mail communication [13], mobile streaming [14], or even ubiquitous connectivity for health monitoring [15]. These works come up with combined techniques that enhance energy-efficiency domain-specifically. For instance, Cheung et al. [13] present a Markov model with states that determine how e-mail synchronization is performed. By performing e-mail synchronization based on user voice activity, the task is performed more energy-efficiently.

*However*, the domain of Web service-based applications has not yet been explored. Thus, the work at hand contributes an analysis of the effect of Web service usage characteristics on energy consumption and an identification of the starting points for extracting rules and models towards energy-efficient development of Web service-based applications.

## III. Web Service Invocation Patterns for Energy Measurements

The purpose of this section is to prepare the SOC-specific analysis of energy consumption. It explains which aspects should be examined in order to design corresponding test cases. The test cases that result from the variation of these aspects will form the basis for defining concrete test cases for the experiments. During the variation of these aspects, it will not be always possible to keep all other parameters constant. However, this complexity will be controlled and considered and the test cases will be designed in a way that allows safe conclusions. In the abstract form in which they appear in this section, the test cases are referred to as *Web service invocation patterns*, or simply *invocation patterns*.

As already mentioned, the standard factors (device and network type, amount of data) will be controlled as needed, or varied to an extent which is known inside the defined invocation patterns. Thus, the question that arises is: "What are the characteristics of Web service call sequences that can be varied along with the standard factors, namely in a way that will allow us to reach new insights?".

Device state changes and "device-network tail times" are critical for energy consumption ([8], [9]). As these are normally determined by time constants [8], an aspect of Web service invocations that can strongly affect them is the timing of the invocations. Further, the logic of service response caching [16] is an aspect of services computing that can implicitly affect standard energy consumption factors, such as the connection establishments [15] or the amounts of transmitted data [8]. Based on the above, but also on findings from own preliminary experiments, it has been concluded that the following features may affect the energy consumption of a given sequence of Web service invocations:

- **Temporal distribution of the invocations:** The lengths of the *time intervals between subsequent invocations* can determine how many times the device changes its state[1], but also how long it remains in idle states, which are important for achieving low energy consumption levels. Generally, it can be assumed that there are three possibilites for the intervals: (i) They are inexistent or so short that a state change is not probable at all, (ii) they have a length that does not allow certainty about the exact state changes between the calls, and (iii) they are so long that the same state changes should normally always occur between subsequent invocations. Therefore, it makes sense to talk about three categorical values for these intervals, namely *short*, *medium*, and *long*. A finer granularity is, of course, thinkable, but is avoided because it does not seem to be necessary and it would complicate the design of the experiment and the drawing of conclusions.
- **Purpose of Web service response caching:** Again because of the device states, but also because of the special costs of consumer-provider connection establishments, the purpose/logic with which Web service response caching is applied (or not), may be a very important factor for energy consumption. As explained in more detail in [17], Web service response caching might be applied with one of two different goals: Either in order to optimize the service invocations (by achieving the transmission of shorter responses) as in [16] and [18], or in order to eliminate them

[1]The number of possible states, as well as their characteristics, may be device-dependent. More detailed discussions about them can be found in [8] and [10]. However, note that the rest of this section handles them abstractly and without loss of generality.

Table I: Combining possibilities for the definition of nine Web service invocation patterns

| | | WS invocation time intervals | | |
|---|---|---|---|---|
| | | *short* | *medium* | *long* |
| **Use of caching** | *No* | InvP1 | InvP2 | InvP3 |
| | *For lightweight messaging* | InvP4 | InvP5 | InvP6 |
| | *For less connections* | InvP7 | InvP8 | InvP9 |

as in traditional caching approaches (e.g., [19]). Both approaches have advantages and disadvantages, e.g., the first ones can guarantee response freshness, while the second ones need less connection establishments. Although both approaches transmit a similar amount of data, the different handling of connection establishments may lead to big differences with regard to energy consumption. Putting it all together, caching may (i) *not be applied at all*, (ii) *be applied in order to achieve more lightweight messaging*, or (iii) *be applied in order to avoid connection establishments*.

The possibility of the existence of further SOC-specific factors cannot be excluded. However, they are expected to have less influence and, due to the controlled experiments, their existence should not affect the findings that will be gained by focussing on the described aspects. Thus, in accordance to the above discussion about the temporal distribution of the invocations and the purpose of Web service response caching, Table I summarizes the nine invocation patterns (InvP1–InvP9) that are obtained by combining all possibilities. The experiments of the next section are designed by controlling the environment and setting appropriate values for the details of the invocation patterns.

## IV. Experiments

The details of the energy measurements and their evaluation are provided in the following. First, the used testbed environment is described. Second, the experimental setup (dependent, independent, and controlled variables) is specified, giving exact values to the invocation pattern details. Finally, selected results are provided and interpreted.

### A. Testbed

Energy measurements have been performed on two devices, running two of the most popular operating systems. In particular, a Symbian-based Nokia E71 device has been used for the main and complete evaluation experiments, while an Android-based Google Nexus One device has been used for validating experiments. All further analyses will be based on the results from the Nokia device, because the latter offers much better and more detailed energy-profiling possibilities, while energy measurements on the Google device have been more coarse and sometimes problematic because of the granularity of the measurements of the corresponding energy-profiling application. However, results from the Google device indicate a similar energy consumption behavior. The same network connections, i.e., the same WLAN and the same UMTS card, have been used for all the experiments.

The software that enabled the measurements consisted of own implementations of Web service-based mobile testbed applications and off-the-shelf energy-profiling applications, which ran in parallel with the testbed applications. The latter have been the Nokia Energy Profiler[2] for the Symbian device and the PowerTutor[3] for the Android device. The testbed applications have been implemented with J2ME and the Android SDK, respectively. Special care has been taken so that the Web service invocations have been the only variable energy-consuming activity during the measurements.

### B. Setup and Test Cases

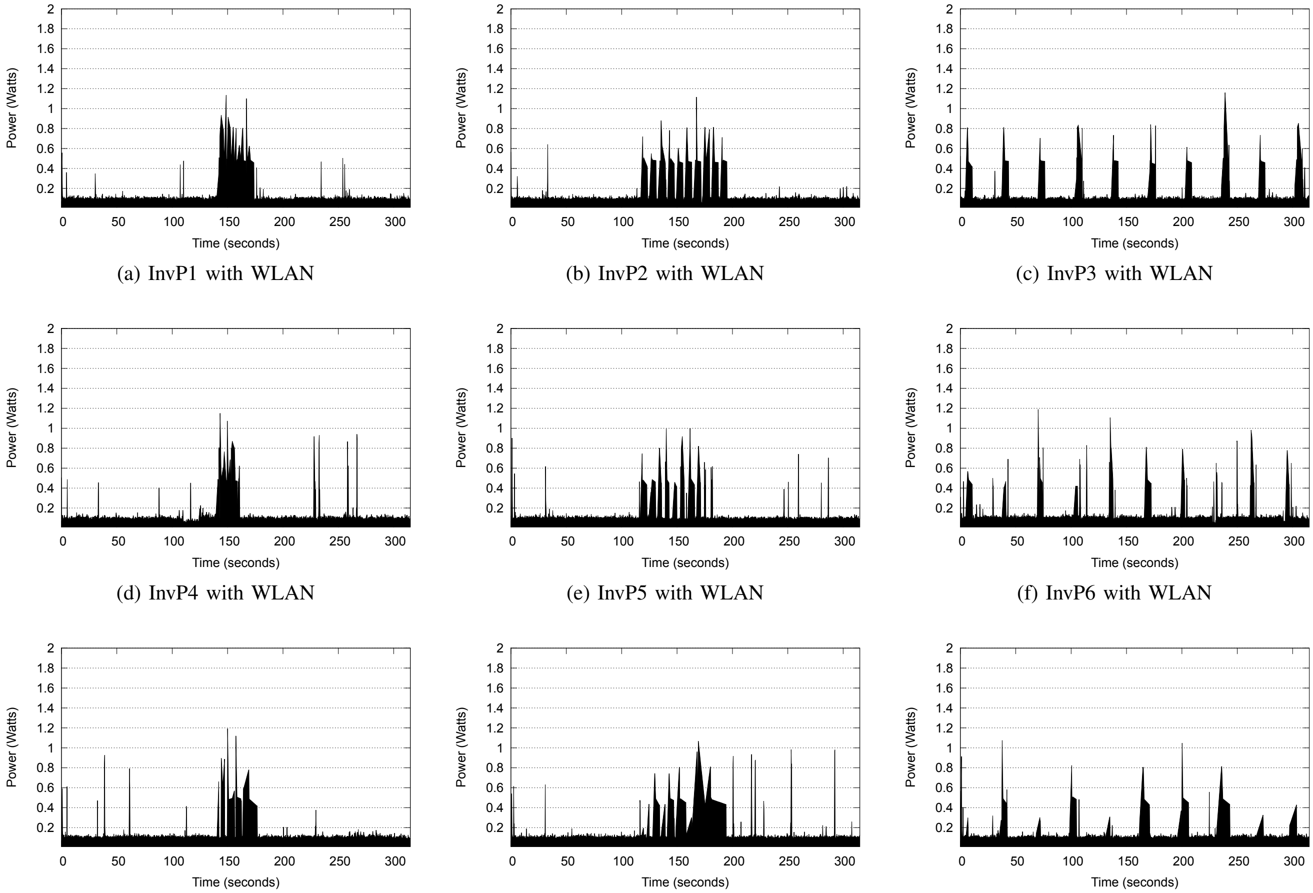
In every single experiment, the same *Web service* has been called the same number of times, namely *ten* times, resulting in responses of the same size, namely ca. 15 kB each time. All single experiments with the same network have had the samed total duration (315 seconds for WLAN and 325 seconds for UMTS, because UMTS sometimes needs this extra time to finish all invocations). Only the invocation patterns, as defined in Section III, have been varied. All energy consumption sources that are not related to the invocation pattern, e.g. display brightness etc. (cf. Section II), have been held constant.

In an attempt to give to the *time intervals* (for the given testbed) the meaning they are meant to have (cf. Section III), the three categorical values have been intuitively interpreted as follows: *short* = 0 seconds (consecutive calls), *medium* = 5 seconds, and *long* = 30 seconds. Experiments with different values could be meaningful and should be part of a more detailed analysis. However, this remains a subject of future work, while the given values reflect the purpose of the definition of the categorical values and have been sufficient for drawing interesting conclusions.

Concerning the use of caching, it has been applied as described in Section III with the following additional setting: For the invocation patterns that involve caching, cache hits with a 50% probability have been assumed. As the approaches imply, in the case of a cache hit, the responses of InvP7-InvP9 were found locally on the device, while the responses of InvP4-InvP6 were SOAP messages with trivial body elements (cf. [18], [16]). Keeping the hit ratio constant at 50% has been sufficient for drawing conclusions, as the effects of caching on the amount of wirelessly transmitted data will be known for all cases and taken into account during the analysis and discussion. Thus, the amount of wirelessly transmitted data is a controlled factor.

[2] http://www.developer.nokia.com/Resources/Tools_and_downloads/Other/Nokia_Energy_Profiler (Last accessed in February 2012).

[3] http://www.powertutor.org (Last accessed in February 2012).




(a) InvP1 with WLAN

(b) InvP2 with WLAN

(c) InvP3 with WLAN

(d) InvP4 with WLAN

(e) InvP5 with WLAN

(f) InvP6 with WLAN

(g) InvP7 with WLAN

(h) InvP8 with WLAN

(i) InvP9 with WLAN

Figure 1: An instance of power consumption over time for each Web service invocation pattern with WLAN

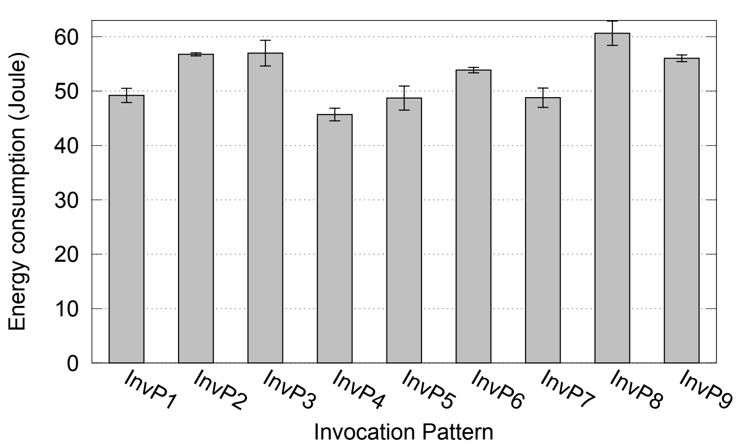
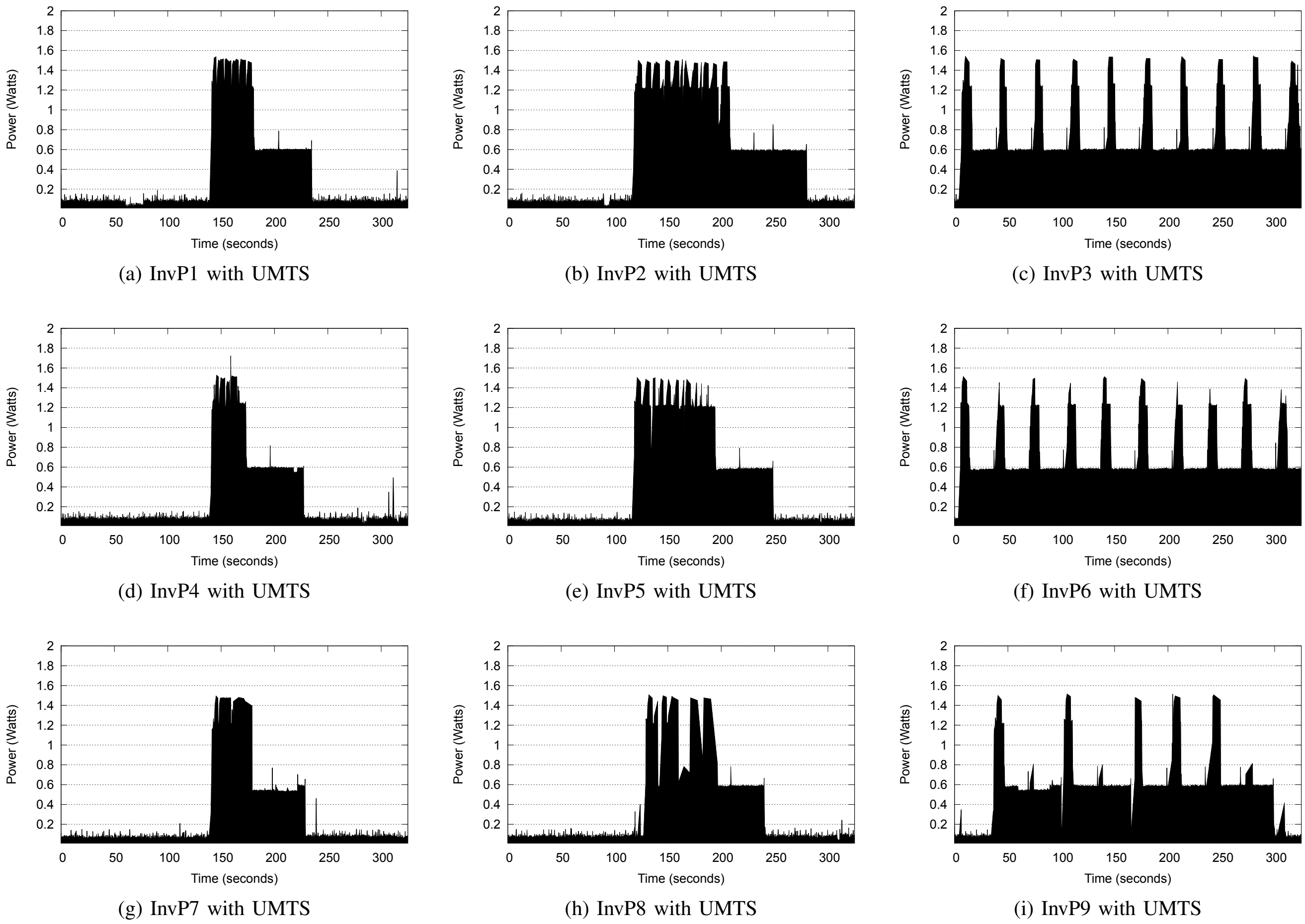



Figure 2: Comparsion of the energy consumption of all Web service invocation patterns for WLAN




(a) InvP1 with UMTS
(b) InvP2 with UMTS
(c) InvP3 with UMTS
(d) InvP4 with UMTS
(e) InvP5 with UMTS
(f) InvP6 with UMTS
(g) InvP7 with UMTS
(h) InvP8 with UMTS
(i) InvP9 with UMTS

Figure 3: An instance of power consumption over time for each Web service invocation pattern with UMTS

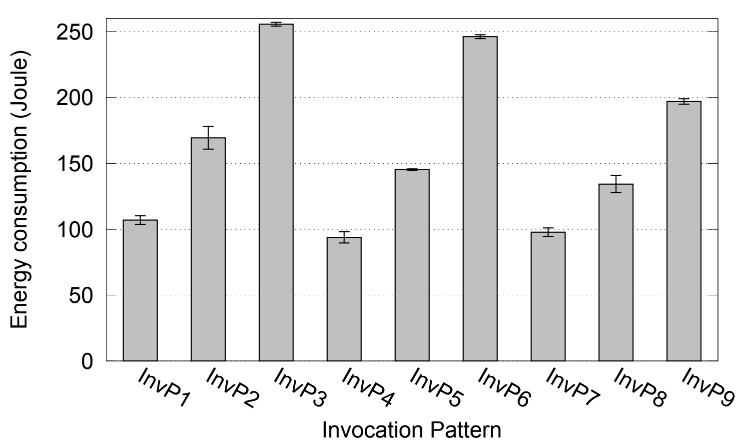



Figure 4: Comparsion of the energy consumption of all Web service invocation patterns for UMTS

### C. Results and Discussion

For illustration purposes, only one representative detailed result of "power consumption over time" is presented for each case, i.e., for each combination of an invocation pattern and an access network. However, each case has been repeated three times, with very small deviations in the results. The values of the average consumed energy of each case are presented with 95% confidence intervals in comparative histograms. Thus, Fig. 1 shows the exact power consumption over time for every invocation pattern with a WLAN connection, while Fig. 2 compares the energy consumption[4] of the invocation patterns. Accordingly, Fig. 3 shows the power consumption over time for every invocation pattern with a UMTS connection, while while Fig. 4 compares the energy consumption of the invocation patterns.

In the following, first, a summary of the results is provided and, second, three important domain-specific findings are explained.

As can be seen, the Web service invocations dominate completely over any side-operation or background-activity. This is indicated both by the clarity with which the occurrence of the invocations can be "read" on the graphs of Fig. 1 and 3 and by the very small deviations among the repetitions of the experiments (cf. confidence intervals in Fig. 2 and 4). The small peaks[5] that appear independently of the invocations in the case of WLAN connections may appear with a different frequency for different WLAN connections. However, their insignificance in this experiment is shown by the small deviations of the total energy consumption.

In general, the invocation patterns InvP1, InvP4, and InvP7 seem to be the most energy-efficient for both WLAN and UMTS, while InvP5 also presents good results in the case of WLAN. A further remark is that, with some exceptions (e.g., InvP8 vs. Invp9), the invocation patterns that perfrom better with a WLAN connection perform better with a UMTS connection, as well. However, the differences are much bigger in UMTS, not only in absolute numbers, but also proportionally. This is a direct consequence of two UMTS characteristics. First, the high power states are higher than those of WLAN although the low ones do not differ. Second, the UMTS tail time "multiplies" the effect of network activity on energy consumption. As a final general remark, it is noted that the results verify the known fact that WLAN is more energy-efficient than UMTS [8], [9], achieving at least 50% (e.g., InvP1, InvP4, InvP7) or even up to 75% (e.g., InvP3) less energy consumption.

A more detailed analysis of the results reveals many differences among the different invocation patterns. Among them, the ones that are most interesting from the point-of-view of services computing and would have not necessarily been expected or foreseen by studying related experiments from other domains, are the following:

- **Clustering of Web service invocations is able to reduce energy consumption:** Although this has been expected for UMTS because of the tail time, the results show how great the effect of the clustering is, achieving from ca. 25% (Inv7 vs. Inv8) up to ca. 60% (Inv1 vs. Inv3) energy reduction. Thus, the handling of subsequent Web service invocations is a very thankful scenario for reducing energy by exploiting UMTS tail times. Interestingly, clustering can often help with WLAN connections, as well, although WLAN connections have no tail time and are thought to depend only on the amount of transmitted data [8]. For example, InvP1 consumes ca. 15% less energy than InvP2 and InvP3. This seems to happen due to the higher temporal overlapping of networking and processing activities when the invocations are clustered. Note, for example, that the percentual energy savings achieved by [12] with CPU- and display-related techniques do not exceed the value of 12%.
- **The number of Web service connection establishments is not always a dominating energy consumption factor:** This remark refers to the comparison of caching for less connections with with caching for lightweight messaging (e.g., InvP4 vs. InvP7, InvP5 vs. InvP8 etc.). This is a very important finding, because the inability of "caching for lightweight messaging" approaches has been assumed to be a big disadvantage concerning energy consumption, e.g., in [17]. However, the very small overlapping of networking and processing activities of InvP3, InvP6, and InvP9, as well as the fact that they have bigger intervals between the connection establishments, prevent them from being more energy-efficient than other invocation patterns. Actually, caching has generally not offered great enhancements (cf., for example InvP4 vs. InvP1), though this could be different for bigger response sizes.
- **For Web service invocations, WLAN energy consumption is not proportional to the amount of transmitted data:** Related work has often come to the conclusion that the consumed energy with WLAN connections is proportional to the size of the data transfer [8]. An argument against this has been already provided in the first remark with regard to clustering. Even stronger evidence appears when caching also comes into play. Compare, for example, InvP1-InvP3 with InvP4-InvP6. Although the first ones have transmitted wirelessly almost twice as much data as the second ones, their energy consumption is only slightly higher. This is again because of the characteristics of temporal overlapping of processing and transmision in the scenario of Web service invocations.

[4] *Energy = power* $\times$ *time*. Thus, energy consumption is calculated as the integral of the curves of power over time, i.e., it is equal to the black-coloured areas of the graphs in Fig. 1 and 3

[5] Because of router beaconing or similar connection maintenance activity.

## V. Towards the Extraction of Rules and Models

This section provides an example of how the presented general experimental findings can be analyzed in more detail for concrete applications or contexts. The example refers to setting adequate time intervals between Web service invocations of a mobile application, when energy-related constraints must be satisfied along with other application-related constraints. Obviously, the procedure could be applied for many different use cases.

Imagine a mobile application developer that must distribute $n$ Web service invocations inside a time period of $t$ seconds[6]. In order to select appropriate time intervals ($i$), the developer can take into account different functions that depend on $i$. Here, for the sake of simplicity, two functions are considered: (i) The percentual energy savings $s(i)$, and (ii) a utility (or satisfaction) function $f(i)$, defined by the developer for the given use case. Function $s(i)$ can be estimated by taking the previous experiments further, enriched with applicaton-specific parameters. This is explained in more detailed in the following paragraphs. Function $f(i)$ may depend on factors such as overloading danger, development overhead, assumed user preferences, and more. In the following, the analysis with which $f(i)$ has been obtained will be omitted, because it is irrelevant and out of scope for the current discussion.

For the estimation of $s(i)$, it is noted that for both WLAN and UMTS, there are values of $i$ for which the energy savings reach a minimum. Further increasing $i$ would not cause higher energy consumptions and, with that, lower values for $s(i)$. For WLAN (and for the given testbed), the results of Section IV indicate that this value is $\leq 5s$, because for $i = 5s$, $s(i)$ seems to have already converged to its minimum value (note that $s(5) \approx s(30)$). Therefore, more fine-grained experiments have been performed (for $i = 0, 1, 2, 3, 4, 5$, and $6s$), providing enough results in order to estimate a function $s_{wlan}(i)$. Similarly, for UMTS, all values of $i$ that exceed the length of the tail time would lead to similar results concerning the consumed energy. Extended measurements and an estimation of the corresponding energy savings function, i.e., ($s_{umts}(i)$), have been performed for UMTS, as well.

Omitting the intermediate results, it is summarized that $s_{wlan}(i)$ had to be split and approximated with the help of a polynomial interpolation (for the first part) and an asymptotic function (for the second part). The approximation of $s_{umts}(i)$ has been simpler, as it could be performed with non-linear regression. The result was a logarithmic function. Concretely, after rounding to three significant digits because

[6]This could be, for example, a social networking application that must update user information from $n$ different sources. Note that the tasks remain the same and only the temporal distribution of the invocations is considered, so that, of course, shorter time intervals do *not* mean more invocations in total.

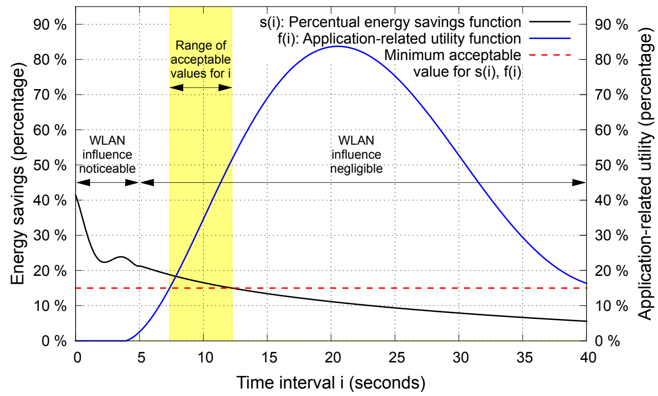



Figure 5: Exemplary exploitation of the energy consumption model for energy-aware mobile Web service-based application development

of the corresponding accuracy of the measurements, the estimated functions are:

$$s_{wlan}(i) = \begin{cases} .195 + .003i^5 - .036i^4 + .158i^3 - .229i^2 - .027i \\ \quad \times 100\%, \quad \text{if } 0 \leq i \leq 5 \\ \\ .092 + .007\frac{i-5}{i-4.6} \\ \quad \times 100\%, \quad \text{if } i > 5 \end{cases}$$

and

$$s_{umts}(i) = .633 - .167 \ln(i+1) \times 100\%$$

Then, based on the assumption that WLAN and UMTS are used in a similar extent[7], the total percentual energy savings function $s(i)$ can be estimated as

$$s(i) = 0.5 \times s_{wlan}(i) + 0.5 \times s_{umts}(i)$$

The estimated functions could be very helpful during application development by themselves. However, Fig. 5 presents $s(i)$ together with an exemplary utility function $f(i)$. Their combined analysis supports an appropriate choice of $i$. Imagine, for example, that the developer sets a requirement of achieving a value of at least 15% for both $s(i)$ and $f(i)$. This minimum requirement is represented in Fig. 5 with the red dashed border line. Then, the yellow-coloured area indicates the range of $i$ for which the requirement is always met.

All in all, with the experimental findings of Section IV as a starting point, models and analyses similar to the one illustrated in Fig. 5 can be designed for various scenarios of mobile Web service-based application development. In addition to time intervals, other parameters mentioned in this paper, such as caching or transmission-processing overlapping, could be considered.

[7]In their experiments with 33 smartphone users, Falaki et al. [20] have also observed similar amounts of traffic for WLAN and UMTS. This assumption can be, anyway, adjusted to each scenario according to the expected application usage.

## VI. Summary and Outlook

This paper has presented experimental results and ideas that can support the reduction of energy consumption of mobile Web service-based applications. By identifying the time intervals between Web service invocations and the logic of Web service response caching as aspects that can lead to interesting domain-specific energy-related findings, the experiments have revealed many interesting details and have concluded with the following three main statements:

First, the temporal clustering of Web service invocations can strongly reduce energy consumption (up to ca. 15% for WLAN and 60% for UMTS). Second, reducing the number of connection establishments between the mobile Web service consumer and the Web service provider through Web service response caching does not have the expected positive effects on energy consumption, because it reduces the temporal overlapping of the tasks of transmission and processing. Third, for Web service invocations, WLAN energy consumption is not proportional to the amount of transmitted data, as has often been assumed in related work.

Finally, it has been shown how the gained knowledge –or the knowledge that can be gained by similar experiments– can be exploited in order to design models, rules, or techniques for energy-aware development of mobile Web service-based applications. An example for identifying appropriate time intervals between Web service invocations application-dependently has been presented, while a more systematic and detailed investigation of further scenarios is a subject of future work.

## Acknowledgements

This work was funded in part by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ09016 (Green Mobility Project). The authors take the responsibility for the contents. Our special thanks goes to Sheip Dargutev and Manuel Zahn for their support.

## References

[1] D. A. Menascé, "QoS Issues in Web Services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72–75, 2002.

[2] M. P. Papazoglou and W.-J. Heuvel, "Service Oriented Architectures: Approaches, Technologies and Research Issues," *The VLDB Journal*, vol. 16, no. 3, pp. 389–415, 2007.

[3] J. Tekli, E. Damiani, R. Chbeir, and G. Gianini, "SOAP Processing Performance and Enhancement," *IEEE Transactions on Services Computing*, vol. PP, no. 99, p. 1, 2011.

[4] C. Frye, "SOA Growth and Change," March 2009, http://searchsoa.techtarget.com/news/1351532/SOA-growth-and-change-TechTarget-survey-shows-SaaS-BPM-emerging, last visited in February 2012.

[5] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast," White Paper, February 2011, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html, last visited in February 2012.

[6] K. Pentikousis, "In Search of Energy-efficient Mobile Networking," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 95–103, 2010.

[7] M. Thirumaran, P. Dhavachelvan, S. Abarna, and G. Aranganayagi, "Architecture for Evaluating Web Service QoS Parameters using Agents," *International Journal of Computer Applications*, vol. 10, no. 4, pp. 15–21, 2010.

[8] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: a Measurement Study and Implications for Network Applications," in *SIGCOMM Internet Measurement Conference (IMC '09)*. ACM, 2009, pp. 280–293.

[9] G. P. Perrucci, F. H. Fitzek, G. Sasso, W. Kellerer, and J. Widmer, "On the Impact of 2G and 3G Network Usage for Mobile Phones' Battery Life," in *European Wireless Conference (EW 2009)*. IEEE, 2009, pp. 255–259.

[10] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," in *IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES/ISSS '10)*. ACM, 2010, pp. 105–114.

[11] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," in *USENIX Annual Technical Conference (USENIXATC'10)*. USENIX Association, 2010, pp. 1–12.

[12] A. Shye, B. Scholbrock, and G. Memik, "Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures," in *Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 42)*. ACM, 2009, pp. 168–178.

[13] T. L. Cheung, K. Okamoto, F. Maker, X. Liu, and V. Akella, "Markov Decision Process (MDP) Framework for Optimizing Software on Mobile Phones," in *ACM International Conference on Embedded Software (EMSOFT '09)*. ACM, 2009, pp. 11–20.

[14] Y. Liu, F. Li, L. Guo, and S. Chen, "A Measurement Study of Resource Utilization in Internet Mobile Streaming," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '11)*. ACM, 2011, pp. 33–38.

[15] A. Rahmati and L. Zhong, "Context-for-Wireless: Context-Sensitive Energy-efficient Wireless Data Transfer," in *International Conference on Mobile Systems, Applications and Services (MobiSys '07)*. ACM, 2007, pp. 165–178.

[16] A. Papageorgiou, M. Schatke, S. Schulte, and R. Steinmetz, "Enhancing the Caching of Web Service Responses on Wireless Clients," in *IEEE International Conference on Web Services (ICWS '11)*. IEEE, 2011, pp. 9–16.

[17] A. Papageorgiou, M. Schatke, S. Schulte, and R. Steinmetz, "Lightweight Wireless Web Service Communication through Enhanced Caching Mechanisms," *International Journal of Web Services Research*, 2012, invited paper under submission.

[18] W. Li, Z. Zhao, K. Qi, J. Fang, and W. Ding, "A Consistency-Preserving Mechanism for Web Services Response Caching," in *IEEE International Conference on Web Services (ICWS '08)*. IEEE, 2008, pp. 683–690.

[19] X. Liu and R. Deters, "An Efficient Dual Caching Strategy for Web Service-Enabled PDAs," in *ACM Symposium on Applied Computing (SAC '07)*. ACM, 2007, pp. 788–794.

[20] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A First Look at Traffic on Smartphones," in *SIGCOMM Internet Measurement Conference (IMC '10)*. ACM, 2010, pp. 281–287.